

A The Standard Rules

A/sr0: *SR0 - Preamble.w* The titling line and rubric, use options and a few other technicalities before the Standard Rules get properly started.

A/sr1: *SR1 - Physical World Model.w* Verbal descriptions of spatial relationships; the hierarchy of kinds of object, and their properties.

A/sr2: *SR2 - Variables and Rulebooks.w* The global variables and those built-in rulebooks which do not belong either to specific actions or to specific activities.

A/sr3: *SR3 - Activities.w* The built-in activities and their default stock of rules; the locale description mechanism.

A/sr4: *SR4 - Actions.w* The standard stock of actions supplied with Inform, along with the rules which define them; and the Understand grammar which corresponds to them.

A/sr5: *SR5 - Phrase Definitions.w* The phrases making up the Inform language, and in terms of which all other phrases and rules are defined; and the final sign-off of the Standard Rules extension, including its minimal documentation.

Purpose

The titling line and rubric, use options and a few other technicalities before the Standard Rules get properly started.

A/sr0.§2 Title; §3-7 Starting up

§1. The Standard Rules are like a boot program for a computer that is starting up: at the beginning, the process is delicate, and the computer needs a fairly exact sequence of things to be done; halfway through, the essential work is done, but the system is still too primitive to be much use, so we begin to create convenient intermediate-level code sitting on top of the basics; so that, by the end, we have a fully flexible machine ready to go in any number of directions. In this commentary, we try to distinguish between what must be done (or else NI will crash, or fail in some other way) and what is done simply as a design decision (to make the Inform language come out the way we want it). Quite interesting hybrid Informs could be built by making different decisions. Still, our design is not entirely free, since it interacts with the I6 template layer (the I7 equivalent of the old I6 library): a really radical alternate Inform would need a different template layer, too.

§2. **Title.** Every Inform 7 extension begins with a standard titling line and a rubric text, and the Standard Rules are no exception:

Version 2/090402 of the Standard Rules by Graham Nelson begins here.

"The Standard Rules, included in every project, define the basic framework of kinds, actions and phrases which make Inform what it is."

§3. **Starting up.** The following block of declarations is actually written by `indoc` and modified each time we alter the documentation. It's a dictionary of symbolic names like `HEADINGS` to `HTML` page leafnames like `doc71`.

...and so on...

§4. Some Inform 7 projects are rather heavy-duty by the expectations of the Inform 6 compiler (which it uses as a code-generator): I6 was written fifteen years earlier, when computers were unimaginably smaller and slower. So many of its default memory settings need to be raised to higher maxima.

Note that the Z-machine cannot accommodate more than 255 verbs, so this is the highest `MAX_VERBS` setting we can safely make here.

Use `MAX_ARRAYS` of 1500.

Use `MAX_CLASSES` of 200.

Use `MAX_VERBS` of 255.

Use `MAX_LABELS` of 10000.

Use `MAX_ZCODE_SIZE` of 50000.

Use `MAX_STATIC_DATA` of 120000.

Use `MAX_PROP_TABLE_SIZE` of 200000.

Use `MAX_INDIV_PROP_TABLE_SIZE` of 20000.

Use `MAX_STACK_SIZE` of 65536.

Use `MAX_SYMBOLS` of 20000.

Use `MAX_EXPRESSION_NODES` of 256.

§5. These, on the other hand, are settings used by the dynamic memory management code, which runs in I6 as part of the template layer. Each setting translates to an I6 constant declaration, with the value chosen being substituted for {N}.

The “dynamic memory allocation” defined here is slightly misleading, in that the memory is only actually consumed in the event that any of the data types needing to use the heap are actually employed in the source text being compiled. (8192 bytes may not sound much these days, but in the tight array space of the Z-machine it’s quite a large commitment, and we want to avoid it whenever possible.)

Use dynamic memory allocation of at least 8192 translates as

```
(- Constant DynamicMemoryAllocation = {N}; -).
```

Use maximum indexed text length of at least 1024 translates as

```
(- Constant IT_MemoryBufferSize = {N}+3; -).
```

Use dynamic memory allocation of at least 8192.

§6. This setting is to do with the Inform parser’s handling of multiple objects.

Use maximum things understood at once of at least 100 translates as

```
(- Constant MATCH_LIST_WORDS = {N}; -).
```

Use maximum things understood at once of at least 100.

§7. Finally, some definitions of miscellaneous options: none are used by default, but all translate into I6 constant definitions if used. (These are constants whose values are used in the I6 library or in the template layer, which is how they have effect.)

Use American dialect translates as (- Constant DIALECT_US; -).

Use the serial comma translates as (- Constant SERIAL_COMMA; -).

Use full-length room descriptions translates as (- Constant I7_LOOKMODE = 2; -).

Use abbreviated room descriptions translates as (- Constant I7_LOOKMODE = 3; -).

Use memory economy translates as (- Constant MEMORY_ECONOMY; -).

Use authorial modesty translates as (- Constant AUTHORIAL_MODESTY; -).

Use no scoring translates as (- Constant NO_SCORING; -).

Use command line echoing translates as (- Constant ECHO_COMMANDS; -).

Use undo prevention translates as (- Constant PREVENT_UNDO; -).

Use predictable randomisation translates as (- Constant FIX_RNG; -).

Use fast route-finding translates as (- Constant FAST_ROUTE_FINDING; -).

Use slow route-finding translates as (- Constant SLOW_ROUTE_FINDING; -).

Use numbered rules translates as (- Constant NUMBERED_RULES; -).

Use telemetry recordings translates as (- Constant TELEMETRY_ON; -).